

1 **"CROSS-MVS SYSTEM SERIALIZED DEVICE CONTROL"**

2

3 **FIELD OF THE INVENTION**

4 The present invention relates to a system for sharing input and output
5 devices, such as tape resources, amongst multiple S/390 systems and their
6 OS/390images. The system incorporates allocation, serialization and locking
7 capabilities so as to manage the shared resources and prevent a device from being
8 allocated to more than one system at a time.

9

10 **BACKGROUND OF THE INVENTION**

11 In mainframe computer installations, independent operating systems
12 (O/S) are operational as multiple images within a single physical hardware structure
13 or system (a box containing one or more CPU's) or a combination of software and
14 multiple boxes. These O/S utilize one or more individual processing units
15 (processors) and share a number of separate physical input/output (I/O) devices,
16 such as robotic tape library, disk drives or peripherals. Sharing of devices permits
17 each O/S to utilize a common peripheral such as the tape drives within a robotic
18 library.

19 This sharing enhances efficiency by utilizing a single device, accessed
20 across a number of O/S. Each O/S is connected and accesses a shared device
21 through a hardware defined communication link or path. In older mainframes, such
22 as IBM System 370 series computers, there is provided upwards of eight paths for

1 as IBM System 370 series computers, there is provided upwards of eight paths for
2 each system. The path took the form of dedicated and multiple physical
3 connections between the box and corresponding ports on the shared device or
4 through multiple addresses on a bus between the control unit and the device. On
5 the system side of the control unit, the path is typically formed of appropriate
6 software messages, e.g. channel control words, carried over a hardware linkage
7 from the system to the control unit. The entire connection from the CPU to the
8 shared device is commonly referred to as a "channel". Each channel is uniquely
9 associated with one shared device, has a unique channel path identifier and has
10 logically separate and distinct facilities for communicating with its attached shared
11 device and the CPU to which that channel is attached. A channel may contain
12 multiple sub-channels, each of which can be used to communicate with a shared
13 device. In this instance, sub-channels are not shared among channels; each sub-
14 channel is associated with only one path. For further details in this regard, the
15 reader is referred to, e.g., page 13-1 of Principles of Operation--IBM System/370
16 Extended Architecture, IBM Publication Number SA22-7085-1, Second Edition,
17 January 1987 (International Business Machines Corporation), which is hereinafter
18 referred to as the "System/370 ESA POP" manual. Hence, commands that
19 emanate from each of these systems, e.g. CPUs travel by way of their associated
20 addressed channels to the shared device for execution thereat with responses, if
21 any, to any such system from the device traveling in an opposite direction. The
22 CPU can also logically connect or disconnect a shared device from a path by

1 issuing an appropriate command, over the path, to the control unit associated with
2 that device.

3 While these physical devices are shared among several different
4 images, each of these devices is nevertheless constrained to execute only one
5 command at a time. Accordingly, several years ago, the art developed a so-called
6 "Reserve/Release" technique to serialize a device across commands issued by a
7 number of different images.

8 One prevalent operating system by IBM is known as the Multiple
9 Virtual Storage image or MVS image. This O/S is also known as MVS/ESA and
10 most currently is OS/390.

11 One successful mechanism for sharing data between multiple
12 systems has been to utilize a coupling facility (CF). A CF is a hardware and
13 software solution for hardwired coupling of multiple systems. The CF links multiple
14 MVS systems, permitting multisystem data sharing and balancing of workloads
15 between and across hardwire-linked systems. Physically, the CF is situated
16 between discrete boxes. CFs are expensive, and are associated with significant
17 overhead to manage what is termed a parallel sysplex.

18 Note that multiple MVS images may exist in a single box or MVS
19 system. Multiple MVS systems are termed a complex.

20 Note that it is often desirable to maintain developer and test MVS
21 systems separate from the production MVS systems, one reason being to avoid
22 potential corruption of the essential production systems during O/S upgrades and

1 application testing. However, even if separate, it would be convenient to be able to
2 access and share tape devices.

3 The problem with sharing across MVS systems stems from having to
4 coordinate the device allocation between images. Without some form of manual or
5 automated management, data integrity is at risk if more than one image tries to
6 access or allocate a tape device at the same time.

7 An operator can manually enter commands to temporarily dedicate a
8 drive to one image prior to allocation but this intervention is labor intensive and
9 prone to errors. The larger the number of images that share these devices, the
10 more difficult it becomes for an operator to manage.

11 Many sites choose to permanently dedicate tape drives to each
12 image. This decision is expensive and can be an inefficient use of tape resources.

13 A single MVS system having multiple jobs, can access devices serially
14 and a manager controls allocation through a common database. Unfortunately,
15 while allocation conflicts for a shared device can be managed successfully on one
16 system, the manager is unaware and unable to manage allocation requests from
17 other images and other systems.

18

SUMMARY OF THE INVENTION

The cross-system tape control ("XTC") of the present invention allows input and output devices that are connected to multiple MVS images, such as tape resources, to be online simultaneously to all systems with physical connectivity to the common resource. Additional features of XTC system include the ability to limit the number of tape drive resources that a particular image can obtain, but not restrict those resources to specific physical devices. A command interface with the operator can display the status of the entire shared tape drive environment from any single system in a XTC complex.

XTC provides an opportunity to make better use of a typically underutilized resource. By making tape drives available to multiple systems simultaneously, it is conceivable that sites should be able to reduce the number of resources required when compared with environments that have different physical resources attached to each system.

Under prior art systems, normally a MVS image within a MVS system can allocate and deallocate without hazard, implicitly knowing that it is the owner of the devices and if it didn't allocate a device then it should be available for it later.

This approach is fine until another MVS system's image tries to make an allocation. Conventionally, not knowing of the existence of other images, the image writing the control database would be able only to update its own allocation information and be insensitive to the allocations of others.

1 The solution is to provide a form of cross-device control or cross-tape
2 control ("XTC") with its ability to allow tape resources that are connected to multiple
3 MVS images to be online simultaneously to all systems. XTC manages the shared
4 resources and prevents the device from being allocated by more than one system
5 at a time.

6 XTC allows the sharing of tape drives, whether round or square, or
7 robotic libraries between multiple parallel SYSPLEXES, multiple logical partitions or
8 LPARS, or multiple LPARS and SYSPLEXES running any mix of MVS/ESA 5.2 and
9 OS/390 operating systems. XTC operates independently of global resource
10 serialization of resources across multiple MVS images.

11 The key is for each system in an XTC complex to monitor each and
12 every other system. If any XTC member becomes busy, blocked otherwise
13 inoperative, the resources owned by the inoperative system can be released by any
14 other operating XTC environment.

15 Some typical scenarios in which XTC could be deployed include
16 utilization of older technology tape drives (such as IBM compatible 3420s) that are
17 used infrequently, but are still required on a number of different systems. Rather
18 than needing to provide physically separate and isolated devices on multiple
19 systems, a smaller number of resources can be shared and used as required.
20 Further, a production environment and a test environment can usefully share a tape
21 resource where the primary user of the tape resource is the production
22 environment. The resources can be shared between the environments without

1 having to vary devices offline and online as required. In current systems, such as a
2 multiple SYSPLEX environment, XTC enables sharing tape drives among systems
3 in different SYSPLEXES and the usual process of IEFAUTOS does not permit
4 device sharing across a SYSPLEX boundary.

5 Simplistically, the above is accomplished by providing a supervisor
6 (XTC) which manages reservation of a device on a system, by an image, and flags
7 it as being in use, regardless of which image on which system reserved it. A
8 common shared control database is required for storing the resource utilization and
9 the reserving image identity. The supervisor intercepts device allocation requests
10 and takes over the reserve/release operation. The control database can be queried
11 by any system at anytime, preferably on a regular heartbeat basis, for information
12 on the availability of a resource or health of a system. The supervisor can perform
13 regular checks on the use of resources, and if a resource is in use by a system that
14 has become non-responsive, the supervisor can release the resource for other
15 images to use.

16 In a broad aspect of the invention a method is provided for cross-
17 system resource sharing of a limited number of serially accessible devices, such as
18 tape drives or printers, which are physically connected in a complex of MVS
19 images, comprising the steps of:

- 20 • providing a control database shared by all MVS images in the
21 complex;

- 1 • periodically monitoring the control database for devices which
- 2 have been allocated and by which image;
- 3 • intercepting a device allocation request from a MVS image;
- 4 • performing a request/release operation on the control database to
- 5 determine if a device or devices satisfy the request;
- 6 • granting allocation of the available devices in the control database
- 7 to the requesting MVS image if the request is satisfied;
- 8 • updating the control database for flagging an allocated device or
- 9 devices as being unavailable, regardless of which image made the
- 10 allocation; and
- 11 • releasing the control database.

12 The above method can be incorporated in a system, preferably
13 operating under OS/390, comprising:

- 14 • a shared control database, preferably a DASD or on a TCP/IP
- 15 network;
- 16 • means for request/release updating operations on the control
- 17 database for flagging which device or devices are unavailable as
- 18 having been allocated by an MVS system and which MVS system
- 19 allocated the device or devices;
- 20 • means for intercepting a device allocation request, preferably
- 21 being a hook such as a subsystem interface function call 78, and

1 which MVS system made the request, using the request/release
2 means for determining if the request can be satisfied from the
3 available device or devices and, if so, satisfying the requests and
4 flagging the allocated devices as unavailable to any MVS system
5 and updating the control database accordingly.

6 7 **BRIEF DESCRIPTION OF THE DRAWINGS**

8 Figure 1 is a flow diagram illustrating the prior art arrangement of
9 systems accessing multiple tape devices;

10 Figure 2 is a flow diagram illustrating an arrangement of systems
11 accessing multiple tape devices utilizing an embodiment of the present invention
12 which uses a shared control database;

13 Figure 3 is a flow diagram illustrating an arrangement of systems
14 accessing multiple tape devices utilizing an embodiment of the present invention
15 which uses a TCP/IP network interface;

16 Figure 4 is a flow chart of one implementation of XTC intercepting an
17 allocation where a device is available;

18 Figure 5 is a flow chart of one implementation of XTC intercepting an
19 allocation where insufficient devices are available; and

20 Figure 6 illustrates code and the results of various status requests
21 made by a user.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Having reference to Fig. 1, a prior art system of connected MVS systems is illustrated, specifically one MVS online system under OS/390 and which is physically connected to devices provided by two IBM or compatible tape drives; a 3590 and a 3490. A batch MVS system, also under OS/390 is connected to the drives. A test MVS system is maintained separately and has no access to the drives.

Having reference to Figs. 2 and 3, a cross-system tape control or XTC is implemented for sharing the resources provided by the two tape drives or robotic libraries. A complex of three MVS systems is illustrated.

In order to share allocation information among all systems that are participating in a given complex, there is a requirement for a control file or database of information accessible or sharable between every system wishing to participate in a particular XTC complex. This shared resource could include a file stored on a Direct Access Storage Device ("DASD") unit (Fig.2) or be one that communicates across a TCP/IP network interface (Fig. 3). The physical devices, resources or tape drives that will be shared must also be physically connected to all systems in a complex.

While the application of the preferred embodiment is typically applied to allocation of tape resources, the invention is equally applicable to any shared device or serially accessed resource such as printers.

1 In the case of a common control database, under shared DASD. XTC
2 must ensure serialized access to the database information. This is managed
3 through a combination of hardware and software file locking. This is also known as
4 request/release or ENQ/DEQ protocol. Such request/release protocol protects
5 common devices in certain phases of multitasking execution or operation.

6 In conventional systems, if a resource is available and an allocation
7 request for a device is received from an image, a System Resource Management
8 (SRM) algorithm, operating under that image, determines which of the one or more
9 devices to allocate to that request. In the case of tape resources, SRM causes a
10 tape to be mounted for use.

11 The hardware lock is used for only a very short period of time at XTC
12 startup to do a validation on whether or not the control database has been
13 initialized. Once that validation has occurred, XTC uses a software lock on the
14 control database under request/release protocols. This technique allows other
15 systems in the XTC complex to read the control database even if they can't
16 currently obtain the database lock. This permits better reporting on which MVS
17 system currently owns the lock in the event that problems arise on the system
18 currently owning the software lock. If an MVS system freezes, its resources can be
19 released from the control database from any active system in the XTC complex and
20 made available to the other systems.

21 Where the common database is available on a TCP/IP network
22 interface, similar issues exist but the data exchange method differs. In this case,

1 the database status information is maintained internally on all systems. A master
2 system maintains ultimate veto authority for any requests. There can be one and
3 only one master system in any given XTC complex. Through network commands, a
4 problem system can be released from the complex from the master system.

5 Having reference to Fig. 4, XTC is a subsystem (dotted boundary)
6 operational under each operating system and is triggered whenever an allocation
7 request is intercepted. XTC maintains the control database and allocates devices
8 as various MVS images allocate them. As one MVS system is unaware of another,
9 XTC performs an allocation of the devices which, being in use and unavailable, may
10 not have been allocated by or to the currently requesting system. Accordingly, one
11 image cannot allocate a device which has been already allocated by itself or
12 another system.

13 XTC utilizes means such as an interface point or special hook for
14 intercepting allocation requests. Basically, the XTC process intercepts every tape
15 allocation request on the MVS image making the request. When an MVS image
16 makes an allocation request, XTC examines the current shared device status to
17 determine if a device of the requested type is available. If the request can be
18 satisfied (i.e. a device or devices of the correct number and type are available), the
19 request is satisfied and MVS allocation is allowed to proceed. XTC updates its
20 control database with flags indicating devices that are being allocated, grants that
21 request and allocates the device or devices. In the control database, XTC writes or
22 flags allocated devices as being unavailable and which MVS system owns them.

1 Having reference to Fig. 5, if an MVS image makes an allocation
2 request for a job requiring one or more devices and insufficient devices are
3 available, or the wrong type of devices are available, then that image's job becomes
4 queued. At some point, when an image deallocates a device, XTC flags the device
5 as available. At the next heartbeat or request/release cycle, those MVS images
6 having jobs in their queues recognize that a device is available and their O/Ss re-
7 drive allocation recovery – the MVS image again makes its allocation request.

8 A global storage table of device allocations, and their owners, is
9 maintained in the control database. Each MVS system is able to access the global
10 table and ascertain the allocation status of devices allocated by other systems.

11 On a regular, periodic cycle, such as on a 2 second timer pop, each
12 MVS image interrogates the control database in the complex. Accordingly, when a
13 device is de-allocated, the control database contents change, a device or devices
14 are flagged as available and the requesting MVS image enters automatic allocation
15 recovery so that the next job pending in the queue can proceed through allocation.

16 To minimize processing overhead, a local storage table of system
17 allocations can be maintained on each MVS system. XTC then enables each MVS
18 image to maintain and perform virtual, background or logical allocations of the other
19 image's allocations as a mirror of the control database status. Therefore an MVS
20 system will be aware that a device is unavailable, even though another system may
21 have claimed it.

1 The means by which XTC is aware that a MVS image is making an
2 allocation request is, in one embodiment, through a hook into the O/S. In most
3 cases, this hook is provided by the subsystem interface (SSI) provided under
4 OS/390. SSI function code (FC) 78 enables one to intercept allocation requests
5 and override the SRM specification. Further, SSI FC 78 permits flagging of the
6 other devices as not being available either.

7 At the simplest level, if an MVS image makes an allocation request,
8 the DASD database file or control database is queried under a typical
9 request/release format. A software lock is applied on the control file, if possible. If
10 the control file is already locked, then this image waits until it can gain control.
11 Predetermined wait thresholds can be set so that a wait duration greater than the
12 threshold would indicate a problem.

13 When the control database becomes free, the control file's device
14 allocated status is cross referenced again with the SSI FC 78 information to
15 determine if a device is available to satisfy the current request. If resource
16 availability is confirmed, then the image claims the resource, writes the new status
17 to the control file and unlocks or releases the software lock.

18 If an MVS image makes an allocation request for $n+1$ resources and
19 only n are available, then the operating system for the image commences an
20 allocation recovery process occurs. If the device is not off-line, and dependent
21 upon a specified allocation algorithm, then the image might wait and hold the
22 device until another/others become free; or the image might wait and not hold the

1 device so that another task, having lesser device demands might use the device in
2 the interim.

3 In the latter no-hold situation, that image will not get any of its
4 requested allocation – simply, the system will not hold up resources for that image if
5 others could use n or less resources. To make the resources available to other
6 less demanding images, an automated unallocation takes place.

7 While the method can be implemented in any shared resource
8 situation, the preferred implementation of XTC is with systems operating under
9 MVS/ESA 5.2 or any release of OS/390 due to those systems having provided
10 convenient subsystem interfaces which enable interception of the allocation
11 requests. Software implementing the system has been tested running in Job Entry
12 Subsystems 2 (JES2) environments.

13 XTC is essentially an extension of the OS/390 operating system. As
14 such, it requires the ability for the console operator to query the subsystem about its
15 current status as well as make requests to update the current environment. XTC
16 builds a console interface component that allows the operator to display the status
17 of the XTC subsystem from a number of different perspectives. For modifiable
18 parameters, XTC will accept requests from the operator for updates to these
19 parameters. The console interface is also very important for recovering a failed XTC
20 environment on another system in the XTC complex. XTC must be able to free
21 resources currently held by another system in the complex if that system has
22 experienced a failure.

1 The operator communication environment in XTC is created by
2 combining components. First of all, MVS must know of the requirement to route,
3 modify and stop requests to the XTC subsystem address space. As well, as part of
4 the subsystem initialization XTC indicates a desire to be able to examine console
5 message traffic and console commands (some of which will be XTC specific).

6 XTC contains a number of features that allow for continuous operation
7 including monitoring of an event notification listener. ENF (Event Notification
8 Facility) is used to recognize when a dynamic change or successful update has
9 been made to the I/O configuration. If new tape devices have been added, the
10 operator can be prompted to include the devices dynamically under XTC control.
11 As well, if devices that are under XTC control have been deleted from the I/O
12 configuration, a decision to reinitialize XTC can also be made. The event
13 notification listener is used to capture successful updates to an OS/390 I/O
14 environment. When XTC recognizes that a successful update has been made to
15 an I/O configuration a special process is triggered. This process examines the
16 contents of the I/O configuration change. If new tape resources have been added
17 to the I/O configuration, XTC will enter an operator dialogue to determine if the
18 resources should be added to XTC control dynamically.

19 If resources have been deleted that are currently under the control of
20 XTC, a console message is issued indicating that a restart of XTC will eventually be
21 required to clean up that condition.

1 When XTC on a given system has gained control of the cross-system
2 resource (either the shared database file lock or the network lock), XTC activity can
3 occur. Five different classes of events could trigger the need to gain control of the
4 environment:

- 5 1. XTC operator command has been entered;
- 6 2. a subsystem event has occurred;
- 7 3. an event notification facility event has occurred;
- 8 4. an IBM robotic tape library allocation request has occurred; or
- 9 5. a heartbeat status event has occurred.

10 Although all the events are important for various reasons, the basic
11 event under XTC management is the device or tape allocation event. This can
12 happen as a result of a type 2 class of event (SSI FC 78 has been invoked) or as a
13 result of a type 4 class event (special exit hook invoked for an IBM robotic tape
14 library allocation under its own Storage Management System (SMS).

15 XTC serializes, through the use of ENQ/DEQ logic, these allocation
16 events. This means that only one allocation event will be actively processed at any
17 point in time. If concurrent events are in process, one event will be active and all
18 others will be queued behind the current active request. This prevents the need to
19 manage the environment in multi-tasking mode and simplifies the code.

20 On startup, XTC is configured for the number and which devices are
21 to be placed under XTC control. XTC then provides the unique capability of being
22 able to logically limit the number of devices that can be concurrently allocated to a
23 given operating system image. These limits are dynamically changeable through

1 the operator interface. This is also a powerful tool in managing resource usage
2 especially in environments where devices may be shared between a very critical
3 production environment and a less critical development or test environment. Also,
4 XTC is able to react dynamically to the addition of new MVS images and devices,
5 without re-initializing, should it change.

6 If an allocation request is re-queued as a result of the unit limit
7 feature, special provisions must be made within XTC to handle what is known as
8 allocation recovery conditions. The operating system of an MVS image will re-drive
9 the allocation request a number of times and XTC must keep track of the status to
10 properly report conditions to the operator.

11 In some cases because of channel path limitations, devices cannot be
12 made online available simultaneously to all OS/390 images that would like access
13 to those devices. For these cases, another class of resource can be placed under
14 XTC control. The resource in this case is the channel itself under Dynamic
15 Channel Reconfiguration (DCR).

16 XTC monitors console message traffic to determine when an event
17 has occurred that may require DCR. XTC then cross-references its table of DCR
18 resources to determine if the current event falls under XTC influence. If this is an
19 XTC eligible event a number of decisions have to be made on both the local system
20 and other systems that could own this same resource. These decisions include:

- 21 • the local system must decide if it currently owns the channel path,
22 but it is simply offline;
- 23 • if not, a request is placed on the cross-system request status;

- 1 • the system owning the requested resource decided if the devices
- 2 on the required channel have more than one channel path
- 3 assigned to them;
- 4 • if so, is at least one other path online and available;
- 5 • if not, a check needs to be made if any of the devices are currently
- 6 allocated;
- 7 • if so, we must wait for all allocations to end; and
- 8 • if not, the channel is eligible to be released.

9 Based on system importance values, provided at system initialization
10 or through the operator interface, XTC then makes a decision to release the
11 channel from the current system. If the channel is released, this information is
12 communicated back to the requesting system through the shared database (either
13 on DASD or through the TCP/IP network).

14 Specifically, running as a subsystem under OS/390 compatible
15 computer systems or complexes, XTC requires less than 4K of common storage
16 below the 16MB line and roughly 200K of common storage above the 16MB line.
17 XTC makes no modification to existing MVS modules.

18 The standard interface points are the subsystem interface (SSI) and
19 the event notification facility (ENF). XTC can run under either the master or JES
20 subsystem and XTC is configured at startup through a parameter dataset that is
21 included in the XTC procedure. The load modules for XTC must reside in an APF
22 authorized library. The inclusion of the XTC procedure, the XTC load modules, and
23 APF authorization can all be done without a system Initial Program Load (IPL) and
24 XTC will dynamically insert the subsystem control blocks it requires if the XTC

1 subsystem name has not been pre-defined in IEFSSN. These capabilities mean
2 that XTC can be installed with no requirement for an IPL.

3 As mentioned earlier, XTC makes no modification to existing MVS
4 modules and the standard interface points are SSI and the event notification facility
5 (ENF). Tape allocation is modified by the SSI FC 78 documented interface, which
6 allows for tape device allocation influence.

7 Several vendors provide robotic tape libraries that can be used by
8 OS/390 operating systems. Devices and libraries, which comply with the generic
9 interface rules, may be controlled through SSI FC 78. However, a robotic library
10 provided by IBM uses Storage Management Subsystem (SMS) to manage the tape
11 devices internal to its library. Tape allocation requests for devices that are SMS
12 managed do not use SSI FC 78 and as a result, device allocation can not be
13 influenced at the same hook point. In other words, this new module does not
14 currently use the subsystem interface as a communication mechanism.
15 Accordingly, a specialized routine, or hook, detects type 4 class events, is applied
16 to capture and influence allocation requests for IBM library devices.

17 Users can also monitor activity and event conditions internal to XTC.
18 The auditing of allocation events occurs if a specific JCL DD exists in the startup
19 procedure for XTC. This log produces a line item entry for each local and cross-
20 system tape allocation event that occurs.

21 The user also chooses to allocate a System Management Facilities
22 (SMF) record number for use by XTC. If an SMF record number is included in the

1 startup parameters for XTC, XTC will capture additional internal event conditions in
2 SMF records. This can be useful information for the customer in tracking usage
3 statistics or for the system administrator if a problem situation occurs. The
4 information can be used for debugging purposes.

5

6 Installation and Operational Control

7 Following is sample startup Job Control Language (JCL) for XTC:

8 //XTC PROC
9 //XTC EXEC PGM=XTCDVR, TIME=1440, DPRTY=(15, 5)
10 //STEPLIB DD DSN=XTC.LINKLIB, DISP=SHR
11 //SHRFILE DD DSN=XTC.SHRFILE, DISP=SHR
12 //PARMLIB DD DSN=SYS1.PARMLIB(XTCPARMS), DISP=SHR
13 //XTCLIB DD DSN=XTC.LINKLIB, DISP=SHR
14 //AUDITLOG DD DSN=XTC.AUDITLOG, DISP=SHR

15 The STEPLIB is optional if the XTCDVR program resides in the
16 system linklist.

17 The SHRFILE represents the XTC control database. It is required.
18 This dataset is a direct access BDAM dataset. The dataset should be set up with
19 DSORG=DA, LRECL=4096, BLKSIZE=4096, KEYLEN=1. A dataset of one or two
20 tracks should be more than adequate for use by XTC.

1 The PARMLIB contains the startup parameters for XTC. A sample
2 set of XTC parameters may look as follows:

```
3           CMDPREFIX=>  
4           SUBSYSNAME=XTC2  
5           UNITLIMIT=32  
6           *TAPEUNIT=ALLTAPE  
7           TAPEUNIT=0380  
8           TAPEUNIT=0381-382  
9           TAPEUNIT=(383,GBL83)  
10          TAPEUNIT=3A0-03A3  
11          3420LIMIT=2  
12          AUTHCODE=XXXXXXXXXXXXXXXXXX
```

13 As can be seen, the XTC subsystem command prefix and subsystem
14 name can be entered through the parameter dataset. There is no default for the
15 command prefix and if no subsystem is provided, XTC will default to a subsystem
16 name of XTC. Limits can be specified for the number of tape drives that can be
17 used by this copy of XTC by using the UNITLIMIT and nnnnLIMIT parameters (for
18 example, where nnnn is either 3420, 3480, 3490, or 3590). You can also specify
19 the devices that XTC is to control. This can be coded in a number of different
20 fashions; using the specific device number, using a range of device numbers, using
21 a device number and a corresponding XTC global name, or indicating that all tape
22 devices are to be controlled by XTC.

23 The XTCLIB DD is required. Even if all XTC load modules are placed
24 in the system linklist, this DD statement must still be coded to reference the library

1 containing the XTC modules. This is the library that XTC uses for all its directed
2 load module loads.

3 The AUDITLOG DD is optional. If XTC is running under the Master
4 (MSTR) subsystem, this DD must use a dataset for output. If you are running
5 under your primary JES, you can use a JES SYSOUT dataset for this DD.

6

7 Operational Commands

8 While XTC is up and running, several commands can be used to
9 obtain information about the current XTC status as well as providing the opportunity
10 to change the current XTC environment.

11 As shown in Fig. 6, system status display commands include the
12 allocation status or on/off line status for a unit, and further, if allocated, who owns it
13 and its job name.

14 Modify commands include:

15 F xtcjname,UNITLIMIT=nn
16 F xtcjname,3420LIMIT=nn
17 F xtcjname,3480LIMIT=nn
18 F xtcjname,3490LIMIT=nn
19 F xtcjname,3590LIMIT=nn
20 F xtcjname,ADDTAPEUNITS=
21 F xtcjname,AUTHCODE=

1 The modify interface also supports the above mentioned display
2 commands. For example:

3 F xtcjname,DISPLAY=UNITS will yield the same result as the
4 stand-alone DISPLAY=UNITS console command.

5 Modifying the unit limits is relatively self evident. Valid values for 'nn'
6 are 0-32.

7 One can also add tape units to XTC through the operator interface.
8 For example, if not all of the tape units were initially included under XTC control in
9 the original start up parameters, use the operator command to add them
10 dynamically.

11 F XTC,ADDTAPEUNITS=0383

12 If your XTC authorization code needs to be replaced, that can be
13 accomplished through the operator interface as well.

14 XTC will also automatically recognize dynamic changes to your I/O
15 configuration that impact tape units. If you dynamically change the I/O
16 configuration to add new tape units, XTC will prompt the operator to find out if the
17 device numbers should be added to XTC control. Conversely, if tape UCB's that
18 are under XTC control have been removed from the I/O configuration, XTC will
19 indicate that an XTC restart should be considered.